2014 14th International Conference on Frontiers in Handwriting Recognition

# A Novel Feature Selection and Extraction Technique for Classification

Kratarth Goel
*Department of Computer Science*
*BITS Pilani Goa*
*Goa, India*
*Email: kratarthgoel@gmail.com*

Raunaq Vohra
*Department of Mathematics*
*BITS Pilani Goa*
*Goa, India*
*Email: ronvohra@gmail.com*

Ainesh Bakshi
*Department of Computer Science*
*Rutgers University*
*New Brunswick, NJ, United States*
*Email: aineshbakshi@gmail.com*

***Abstract*—Pattern recognition is a vast field which has seen significant advances over the years. As the datasets under consideration grow larger and more comprehensive, using efficient techniques to process them becomes increasingly important. We present a versatile technique for the purpose of feature selection and extraction - Class Dependent Features (CDFs). CDFs identify the features innate to a class and extract them accordingly. The features thus extracted are relevant to the entire class and not just to the individual data item. This paper focuses on using CDFs to improve the accuracy of classification and at the same time control computational expense by tackling the curse of dimensionality. In order to demonstrate the generality of this technique, it is applied to two problem statements which have very little in common with each other - handwritten digit recognition and text categorization. It is found that for both problem statements, the accuracy is comparable to state-of-the-art results and the speed of the operation is considerably greater. Results are presented for Reuters-21578 and Web-KB datasets relating to text categorization and the MNIST and USPS datasets for handwritten digit recognition.**

***Keywords*-MNIST; USPS; Reuters-21578; WebKB; Handwritten Digit Recognition; Text Categorization; SVM; Pattern Recognition**

## I. Introduction

The field of pattern recognition is one that is broad and rapidly advancing. Classification tasks find application in a myriad of real-world situations. Over the past several years, classification problems have seen a multitude of approaches with increasing sophistication. While the results obtained in this manner are impressive, they often require extremely high computation time which can be crippling or impossible to implement for independent researchers on mainstream computers. Addressing the curse of dimensionality also becomes imperative as datasets grow tremendously.

Considering the diversity in problem statements that fall under the broad scope of pattern recognition, it would be greatly beneficial to solve them successfully using a generic technique that could address the difficulties associated with most approaches.

Handwritten digit recognition is the process of receiving and correctly interpreting a legible hand-drawn digit from an input source (paper or photographs) by comparing it with previously trained data. Text categorization is the process of classifying documents into one out of a set of predefined labels. While the former suffers from difficulties arising due to effective feature selection, the latter is primarily afflicted by the curse of dimensionality. Considering the immense difference in the usual approach to solve these problems, a technique that can effectively tackle both of them is greatly desirable.

With this end in mind, we introduce a novel and versatile technique for feature selection and extraction to a variety of classification models and demonstrate its successful application to the problem of handwritten digit recognition and text categorization. The features obtained, called Class Dependent Features (CDFs), are inherent to a particular class label and are extracted accordingly, unlike most popular techniques which consider each individual data item separately. Feature vectors thus formed are then provided to a classifier which performs the classification task using an SVM.

The organization of the paper is as follows: Section (2) describes the work done on both problem statements, Section (3) describes, in detail, the working of our technique for feature selection and dimensionality reduction and includes a brief discussion on points regarding its implementation. Results are tabulated in Section (4) using the MNIST [11] and USPS [9] datasets for handwritten digit recognition and the WebKB [4] and Reuters-21578 [12]. Section (5) discusses possible future applications of this technique and concludes the paper.

## II. Related Work

The field of digit recognition has seen extensive research over the last fifteen years, and the results have been increasingly promising. Since there are thousands of images in a typical training or testing database, suitable feature selection and extraction [15] is a significant issue.

Belongie *et al.* [1] used the technique of shape context matching with K-nearest-neighbours which produced results comparable to the current state-of-the-art. The work done by Kegl and Busa-Fekete [8] on boosting products of base classifiers with Haar features has also shown some very competitive results. LeCun *et al.* introduced the concept

2167-6445/14 $31.00 © 2014 IEEE
DOI 10.1109/ICFHR.2014.25

of Convolutional Neural Networks and produced excellent results with the LeNet-5 classifier. Deng and Yu [7] worked with deep convex nets and produced the current best result (0.83% error rate) on the MNIST dataset with a generic algorithm *i.e.* an algorithm which does not make explicit use of the fact that the vector matrices represent images. It must be noted though that all of these take considerably long to train and have very high time complexity, making them very difficult to implement for mainstream applications.

DeCoste and Scholkopf [6] introduced the technique of using a Virtual-SVM (V-SVM). However, this is a semi-generic algorithm (it makes use of the fact that the vector matrices represent images, but not necessarily characters), as it jitters the image by one or more pixels in each direction to create new virtual support vectors. This also brings about an increase in the time complexity, as the number of support vectors increases by a large amount.

In the case of text categorization, a literature survery reveals a fairly standard set of techniques that are followed, as outlined here. Regardless of the learning algorithm, text classification is a challenging problem since the dimensionality of the data is very high. Due to this reason, feature selection is a fundamental issue in text classification problems and consists of two main steps: pre-processing and classifier training.

There are numerous studies on feature selection which evaluate and compare most of the popular feature selection metrics [16] [10] such as Information Gain, Chi-square statistics etc. Ozgur and Gungor [13] analyses two keyword selection policies named as class- and corpus-based keyword selection by using SVMs on datasets of different skewness and sizes.

The Tf-Idf statistic [5] increases proportionally with the number of times a word appears in the document, but is offset by the frequency of the word in the corpus. It inadvertently decreases the statistic for words innate to an entire class label for a corpus where a majority of the documents belong to that class. Words that occur repeatedly in a particular class label are either commonly used words in the English language or words that collectively describe that class. The IDF factor for the latter reduces drastically and these words tend to be ignored thereby resulting in poor features and lower accuracy for classification.

### A. Our Contribution

In this techique, we focus on two problems at opposite ends of the spectrum of pattern recognition - one which focuses on feature selection and one which focuses on reducing dimensionality.

We propose a novel and robust technique for feature selection and extraction which gives results comparable to the current state-of-the-art with the added advantage of being very fast and easy to implement on a range of devices. The algorithm works by first selecting features relevant to their class label and extracts them accordingly. These extracted features are interesting because they are relevant to the entire class they are a part of, not just the individual data item they are extracted from. We call these features *Class Dependent Features* (CDFs). Moreover the entire learning problem is then broken down into smaller classification tasks by creating a SVM for each pair of class labels. Each pair of class labels has a varying number of feature vectors which enables intricate parameter selection for each classifier thus enabling improved learning of the pair of class labels as opposed to selecting universal parameters for all classes. However the formation of the pair of class labels varies depending on *one-vs-one* or *one-vs-all* classifiers. In the former, a pair consists of two class labels juxtaposed with each other and $\binom{n}{2}$ classifiers are formed, assuming $n$ to be the total number of class labels. In the latter, however, a pair consists of a particular class label juxtaposed with the rest of the data resulting in $n$ classifiers being formed. The CDFs are then passed to the classifier to complete the task.

## III. The Algorithm

In this section, we explain the algorithm we use to generate CDFs to train and test our data. For the purposes of the experiments conducted using this technique, we assume that the datasets under consideration are presented in a form that can be used for arithmetic manipulation (*e.g.* intensity values of pixels for the MNIST and USPS datasets; word count of the stemmed and vectorized documents in the case of Reuters-21578 and WebKB datasets).

### A. Feature Selection

We first create a measure *T* for each class label. This is done by aggregating corresponding feature values of every member of each class label. For example, in the case of the MNIST and USPS datasets, the class labels are the ten possible digit classes $(0, 1, 2, ..., 9)$. The members of each label are the images of each handwritten digit and the feature values are thus the pixel intensities. So the essence of what is to be done is to average the pixels in the $i^{\text{th}}$ row and $j^{\text{th}}$ column of all the images in a particular class label, to create one element of the measure $T$. Therefore the dimensionality of $T$ will be the same as the dimensionality of each image in the dataset.

Consider a vector $P = \{P_1, P_2, ...., P_m\}$ representing the set of all class labels in the training dataset ($m$ is the total number of class labels in the dataset). Each $P_c = \{p_1, p_2, ...., p_M\}, p_k \in \mathbb{R}^N, \forall k \in [1, M]$ further represents all data points in the $c^{\text{th}}$ class label.[1] Then the function $f(P_c) = \{a_{c_1}, a_{c_2}, ..., a_{c_N}\}$ representing the *summation function* for the $c^{\text{th}}$ class label is given by

[1]Here, the data points refer to individual components of a class label, *i.e.* the individual images (composed of pixel intensity values) in the MNIST and USPS datasets or documents (composed of words) in the Reuters-21578 and WebKB datasets.

$$a_{ci} = \sum_{k=1}^{M} p_k(i) \,. \tag{1}$$

where $N$ denotes the dimensionality of $p_k$, $M$ is the cardinality of the $c^{\text{th}}$ class label and $a_{ci} \in \mathbb{R}$. Here, for instance, the element $p_k(i)$ would be the pixel intensity value of the $i^{\text{th}}$ pixel of the $k^{\text{th}}$ image, belonging to the class label $c$ (say, the digit 0) of the MNIST dataset.

We then obtain the measure $T(P_c) = \{q_{c_1}, q_{c_2}, ...q_{c_N}\}$

$$q_{ci} = a_{ci}/M \tag{2}$$

$\forall i \in [1, N]$.

Figure 1 shows the generated measure *T* for the MNIST dataset, while the measure generated for the USPS dataset is shown in Figure 2.

Now that we have the measure *T* for each class label, we must establish a relation $\mathbf{R_{xy}}$ between each pair of class labels $P_x$ and $P_y$. This is used to obtain the degree of relatedness between the two class labels $P_x$ and $P_y$, which gives us the degree of similarity and dissimilarity between them. The experiments in this paper were conducted taking $\mathbf{R_{xy}}$ as

$$\mathbf{R_{xy}} = \{q_{xi}/q_{yi} \mid \forall q_{xi} \in T(P_x) \; and \; \forall q_{yi} \in T(P_y)\} \,. \tag{3}$$

We then take the mean of all the values of $q_{xi}/q_{yi} \in \mathbf{R_{xy}}$ as shown below.

$$\mu_{xy} = \frac{\sum_{i=1}^{N}(q_{xi}/q_{yi})}{N} \,. \tag{4}$$

We generate two thresholds $\tau$ and $\tau'$, given by

$$\begin{aligned} \tau &= b\mu_{xy} \,. \\ \tau' &= b'\mu_{yx} \,. \end{aligned} \tag{5}$$

where $b, b' \in \mathbb{R}$

Values of $q_{ci} \in T(P_c)$ greater than the thresholds $\tau$ or $\tau'$ will be selected as the feature locations for the $c^{\text{th}}$ class label. Considering $T(P_c)$ as a vector of real values, only those indices *i.e.* $i$'s in $T(P_c)$ for which $q_{ci}$'s have their values greater than either threshold are chosen as class dependent.

For instance, only the pixel intensity values of the pixels belonging to an image in the $c^{\text{th}}$ class label (say, the digit 0) of the MNIST dataset located at the indices selected by the thresholding procedure would be selected as the Class Dependent Features (CDFs) for the purpose of classification of this particular image. Hence the values $b$ and $b'$ can be thought of as parameters controlling dimensionality of the input space for the given problem statement.

Thus we define a new dataset $P' = \{P'_1, P'_2, ...., P'_m\}$, where each $P'_c$ is the set of modified data items in the $c^{\text{th}}$ class label, given by

$$P'_c = \{p'_1, p'_2, ...., p'_M\} \,. \tag{6}$$

where each $p'_k, \forall k \in [1, M]$ is,

$$p'_k(i) = \begin{cases} p_k(i), & if \; q_{ci} > \tau \; or \; q_{ci} > \tau' \\ NULL, & otherwise \end{cases} \tag{7}$$

$\forall i \in [1, N]$.

Thus we have effectively reduced the dimensionality of the dataset by keeping only the class dependent values of these features *i.e* the values of the features greater than the thresholds $\tau$ or $\tau'$. This can be interpreted as the non-NULL values represented in equ. (7).

### *B. Feature Extraction*

With our thresholds $\tau$ and $\tau'$ calculated, we can now proceed with the extraction of class dependent features (CDFs) for the pair of class labels $x$ and $y$. In order to do so, we use the concept of the *Kullback-Leibler (KL) divergence*.

The KL divergence is a non-symmetric measure of the difference between two probability distributions $P$ and $Q$. Specifically, the KL divergence of $Q$ from $P$, denoted $D_{KL}(P \parallel Q)$, is a measure of the information lost when $Q$ is used to approximate $P$.

The feature vector $\mathbf{F_{xy}}$ used for the purpose of classification between the pair of class labels $x$ and $y$ is calculated as follows:

$$\mathbf{F_{xy}}(k) = D_{KL}(p'_k \parallel T(P_x)) \,. \tag{8}$$

Also a set of labels is created to be given to the classifier as follows:

$$\mathbf{L_{xy}}(k) = \begin{cases} 1 & p'_k \in P'_x \,. \\ -1 & p'_k \in P'_y \,. \end{cases} \tag{9}$$

We have now obtained our CDFs for each class label comparison. These features will be passed to the classifier for training on the dataset. This process is repeated for all pairs of class labels $x$ and $y$ to cover the entire dataset.

This discussion is outlined for the *one-vs-one* paradigm of classification. The *one-vs-all* paradigm can just as easily be used; here, the relation $\mathbf{R_{xx'}}$ will be used instead of $\mathbf{R_{xy}}$. $\mathbf{R_{xx'}}$ is defined as

$$\mathbf{R_{xx'}} = \{q_{xi}/q_{x'i} | \forall q_i \in T(P_x) and \forall q_{x'i} \in T(P_{x'})\}. \tag{10}$$

where $T(P_{x'})$ is given by,

$$T(P_{x'}) = \sum_{j=1, j\neq x}^{m} T(P_j(i)) \,. \tag{11}$$

The choice between the *one-vs-one* ($\binom{n}{2}$ classifiers) and *one-vs-all* ($n$ classifiers) paradigms is not a constraint placed by our technique; it is determined solely by the needs of the problem statement. For instance, to tackle the problem of Handwritten Digit Recognition we have used the *one-vs-one* paradigm and for Text Categorization we use the *one-vs-all* paradigm.

Figure 1: Measure *T* of images of all class labels in the MNIST dataset

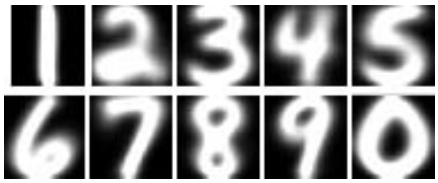

Figure 2: Measure *T* of images of all class labels in the USPS dataset

### C. Discussion

The formation of the measure *T* for each class label is done primarily to obtain a unit that may stand for the entire class label, *i.e.* the measure *T* for each class label can be thought of as a representation of the entire class label. The next step is to establish the degree of similarity and dissimilarity between class labels, which will subsequently lead to appropriate feature selection. To this end, per unit division becomes a particularly apt choice for the relation $\mathbf{R_{xy}}$. Division between two real numbers (say $h$ and $k$, $k \neq 0$) can result in one of three possible situations - either their ratio is significantly greater than or less than unity or it approximately equals unity. The first two cases suggest a great dissimilarity between $h$ and $k$ while the third case indicates a high degree of similarity between them. This argument implies that the two thresholds $\tau$ and $\tau'$ that are generated are thus a means of isolating the dissimilar elements of the measure *T* from the similar elements. This distinction helps in distinguishing between class labels and culminates in the selection of relevant features.

The relations formed in the data using our technique can be looked at from both a macroscopic and microscopic viewpoint - the degree of relatedness between two class labels (macroscopic) adds a level of understanding to the data, which can be used to calculate the features of the individual data point in its class label (microscopic). $\mathbf{R_{xy}}$ gives us the correspondence between each pair of class labels by identifying the points in the input space that contribute actively to determine the class label. This correspondence can be considered as an added layer of learning that further improves classification accuracy. So the algorithm not only learns the features by giving them to a classifier but also - by just looking at the data - learns what features are relevant.

## IV. Experimental Results

In order to test our technique and prove its generality and versatility, we used it on two fundamentally different problem statements - handwritten digit recognition and text categorization. For the former, we used the well-known MNIST and USPS datasets and used the WebKB and Reuters-21578 datasets for the latter. We used SVMs as the classifier for both problem statements and the optimum parameters were determined by using $n$-fold cross-validation.

### A. Handwritten Digit Recognition

As can be observed from Table I, our technique produces the best reported error rate among generic classification algorithms (no preprocessing was carried out on the data) after Hinton and Salakhutdinov's (2007) deep belief nets (1.00%) [14] and Deng and Yu's deep convex nets (0.83%). It outperforms Kegl and Busa-Fekete's products of boosting stumps (1.26%).

We now turn our attention to the performance of CDFs with the USPS dataset, the results of which are presented in Table II.

Table I: Comparison of various generic techniques with error rates (%) on the MNIST dataset.

| **Techniques** | **Error** |
|---|---|
| Linear classifier (1-layer NN) | 12.0 |
| K-nearest-neighbors, L3 | 2.83 |
| Products of boosted stumps | 1.26 |
| 40 PCA + Quadratic classifier | 3.3 |
| SVM, Gaussian kernel | 1.4 |
| 3-layer NN, 500+150 hidden units | 2.95 |
| 2-layer NN, 800 HU, Cross-Entropy | 1.53 |
| Deep Belief Net | 1.0 |
| Deep Convex Net | 0.83 |
| Large Convolutional Net (no distortions) | 0.62 |
| **CDFs, SVM, 2-degree poly kernel** | **1.25** |

Table II: Comparison of various techniques with error rates (%) on the USPS dataset.

| **Techniques** | **Error** |
|---|---|
| Relevance vector machine | 5.1 |
| Convolutional Neural Net (LeNet-1) | 5.0 |
| Kernel densities, virtual data | 4.2 |
| Products of boosted stumps | 4.24 |
| SVM (raw pixels) | 4.0 |
| LDA, virt. data, Gauss. mix. density. | 3.4 |
| Two-sided tangent distance | 3.0 |
| 3-NN, 2-D deformation model | 2.7 |
| Preprocessing, SVM | 2.5 |
| **CDFs, SVM, 2-degree poly kernel** | **4.78** |

### B. Text Categorization

To evaluate the utility of the various feature selection methods used, the F1-measure is used which combines

precision and recall, two commonly used measures of text categorization performance.

Precision is defined as the ratio of correct classification of documents into categories to the total number of attempted classifications. Recall is defined as the ratio of correct classifications of documents into categories to the total number of labeled data in the testing set. For multi-label classification, they are formulated as follows:

$$Precision(\pi) = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} TP_i + \sum_{i=1}^{n} FP_i} \quad (12)$$

$$Recall(\rho) = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} TP_i + \sum_{i=1}^{n} FN_i} \quad (13)$$

Where $TP_i$ , $FP_i$ and $FN_i$ is the number of True Positives, False Positives and False Negatives respectively.

There are two different ways to calculate the F1 statistic namely Micro- and Macro-averaged F1-measures. The former reflects the overall accuracy better, while the latter is good at measuring the performance of the classifier on rare categories since it gives equal weight to all classes regardless of the frequency of the class.

$$F(micro - averaged) = \frac{2\pi\rho}{\pi + \rho} \quad (14)$$

$$F(macro - averaged) = \frac{\sum_{i=1}^{n} \frac{2\pi\rho}{\pi+\rho}}{n} \quad (15)$$

Hence a complete overview of the classification task can be obtained by viewing and comparing these two statistics. Table III and Table IV show precision, recall and F measure for each label in the corpus as well as the micro and macro averaged F value for each set. These results are obtained using a SVM classifier for the Reuters-21578 Dataset and WebKB Dataset respectively. The average number of features can be controlled by varying the threshold and the resulting Micro F1 scores are shown in Figure 3.

In order to benchmark our feature selection and extraction technique, it is compared with contemporary feature selection techniques such as Information Gain, Expected Cross Entropy, Mutual Information, Odds Ratio, the Weight of Evidence of Text, CHI and Gini index on the Reuters Dataset. The SVM classifier is again used for all techniques thereby maintaining uniformity and the results for the Reuters-21578 dataset are tabulated in Table V. It is evident that CDFs outperform all other techniques and are a significant improvement upon them. The macro and micro F scores (89.28% and 96.32%) exceed Gini index, the second best, by 20% and 6.5% respectively.

On an absolute scale, our results are on par with state-of-the-art results on the same datasets. While there are methods which outperform our classifier, we would like to point out that all of them either use complex, image-based techniques or are algorithms with extremely high computational complexity. Some of them create and use virtual data, which also significantly increases computation time. This means that these algorithms are essentially limited in their application to only machines with extremely sophisticated hardware which are capable of running them. Our technique, on the other hand, has no such limitations and can give strong results in a small amount of time. This makes its range of application much broader.

Table III: Precision Recall and F1 Statistics on Reuters-21578 Dataset.

| Category | Precision | Recall | F1 |
|---|---|---|---|
| EARN | 98.36 | 99.54 | 98.94 |
| ACQ | 96.30 | 97.27 | 96.79 |
| CRUDE | 96.52 | 91.74 | 94.07 |
| TRADE | 89.88 | 94.66 | 92.20 |
| MONEY-fx | 83.53 | 81.61 | 82.56 |
| INTEREST | 92.88 | 80.27 | 86.10 |
| SHIP | 90.00 | 75.00 | 81.82 |
| GRAIN | 75.00 | 90.00 | 81.82 |
| **MICRO AVG F1** | **96.32** | **MACRO AVG F1** | **89.29** |

Table IV: Precision Recall and F1 Statistics on WebKB Dataset

| Category | Precision | Recall | F1 |
|---|---|---|---|
| COURSE | 93.58 | 89.35 | 91.42 |
| FACULTY | 91.35 | 84.75 | 87.93 |
| PROJECT | 76.07 | 73.80 | 74.92 |
| STUDENT | 88.02 | 87.86 | 87.94 |
| **MICRO AVG F1** | **87.14** | **MACRO AVG F1** | **85.55** |

Table V: Benchmarking CDFs with contemporary techniques on Reuters-21578 using SVM classifiers.

| Feature Weight Function | Macro F | Micro F |
|---|---|---|
| TFxIDF | 62.63 | 84.73 |
| TFxGINI | 69.82 | 89.79 |
| TFxIG | 63.88 | 84.64 |
| TFxCROSS-ENTROPY | 66.63 | 86.55 |
| TFxX2 | 60.88 | 83.71 |
| TFxMUTUAL-INFO | 68.15 | 87.59 |
| TFxODDS-RATIO | 69.16 | 88.05 |
| TFxWEIGHT OF EVID | 64.24 | 85.22 |
| **CDF** | **89.29** | **96.32** |

These experiments were carried out on a 2nd Generation Intel Core i5-2410M processor running Ubuntu 13.04 and the code was implemented using OpenCV 2.4.6.1 [3]. The training and testing on the USPS dataset takes 6.37 seconds while on the MNIST dataset it takes 5.77 minutes. In comparison, while running the MNIST task using Stacked

Denoising Autoencoders, pre-training takes 585.01 minutes. Fine-tuning is completed after 36 epochs in 444.2 minutes. The final testing score obtained is 1.3%, which our technique outperforms by 0.05%. These results were obtained using Theano 0.6rc3 [2] on a machine with an Intel Xeon E5430 @ 2.66GHz CPU, with a single-threaded GotoBLAS. This serves to illustrate the accuracy our technique brings with its computation time being a fraction of that taken by Stacked Denoising Autoencoders.

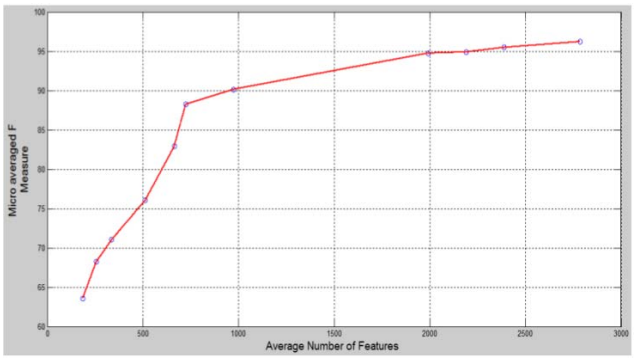


Figure 3: Graph depecting variation of Micro Avg F Measure w.r.t. Average number of CDFs on Reuters-21578 Dataset

## V. Conclusions and Future Work

We have proposed a technique for feature selection and extraction using *Class Dependent Features* (CDFs). Our technique has been tested on the MNIST and USPS datasets for handwritten digit recognition as well as on the Reuters-21578 and WebKB datasets for text categorization. We have obtained strong competitive results using an SVM with a degree-2 polynomial kernel and these results along with those of comtemporary techniques have been tabulated. Our results are comparable to the current state-of-the-art, and on par with the current best generic algorithms.

We are currently working on improving the results quoted in this paper and are looking towards applying our technique on other problem statements such as document summarization and analysis of EEG signals for human-computer interaction.

## References


[1] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(24):509–522, 2002.

[2] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio. Theano: a CPU and GPU math expression compiler. *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010.

[3] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[4] M. Craven, D. DiPasquo, D. Freitag, A. McCailum, T. Mitchell, K. Nigam, and S. Slattery. Learning to extract symbolic knowledge from the world wide web. *Proc. of AAAI*, pages 74–81, 1998.

[5] F. Debole and F. Sebastiani. Supervised term weighting for automated text categorization. *SAC*, pages 784–788, 2003.

[6] D. DeCoste and B. Scholkopf. Training invariant support vector machines. *Machine Learning*, 46:161–190, 2002.

[7] L. Deng and D. Yu. Deep convex net: A scalable architecture for speech pattern recognition. *INTERSPEECH*, pages 2285–2288, 2011.

[8] B. Kegl and R. Busa-Fekete. Boosting products of base classifiers. *Proceedings of the 26th International Conference on Machine Learning*, 2004.

[9] D. Keysers. Usps dataset. 1994. Available: http://www-i6.informatik.rwth-aachen.de/ keysers/usps.html.

[10] F. S. L. Galavotti and M. Simi. Feature selection and negative evidence in automated text categorization. *Proceedings of the ACM/SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2000.

[11] Y. LeCun, C. Cortes, and C. Burges. The mnist database of handwritten digits. 1995. Available: http://yann.lecun.com/exdb/mnist/index.html.

[12] D. Lewis. Reuters-21578 text categorization test collection. 2004. Available: http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html.

[13] A. Ozgur and T. Gungor. A comparative study on feature selection in text categorization. *Proceedings of the International Conference on Machine Learning*, pages 412–420, 1997.

[14] R. Salakhutdinov and G. Hinton. Learning a nonlinear embedding by preserving class neighbourhood structure. *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, pages 412–419, 2007.

[15] P. Simard, D. Steinkraus, and J. Platt. Best practices for convolutional neural networks applied to visual document analysis. *IEEE Proceedings of the 7th International Conference on Document Analysis and Recognition*, pages 958–962, 2003.

[16] Y. Yang and J. O. Pederson. A comparative study on feature selection in text categorization. *Proceedings of the International Conference on Machine Learning*, pages 412–420, 1997.